

# 5954025 - Sistemas Distribuídos

## Aula 11b - Algoritmos de Eleição

**Prof. Dr. Denis M. L. Martins**

DCM | FFCLRP | USP

# Objetivos de Aprendizagem

- Explicar por que sistemas distribuídos precisam de mecanismos de eleição.
- Identificar situações em que a falha de um coordenador compromete o sistema.
- Descrever o funcionamento do algoritmo **Bully**.
- Descrever o funcionamento do algoritmo **Ring**.
- Comparar os algoritmos quanto a mensagens, simplicidade e tolerância a falhas.
- Resolver exercícios conceituais e práticos sobre eleição de líderes.

# O Problema da Coordenação

Em um sistema distribuído, vários processos executam em máquinas diferentes e se comunicam por mensagens. Em muitos casos, um processo precisa assumir o papel de **coordenador** ou **líder**.

- Esse coordenador pode ser responsável por:
  - organizar o acesso a um recurso compartilhado;
  - iniciar tarefas globais;
  - controlar réplicas;
  - coordenar transações;
  - tomar decisões em nome do grupo.
- **Exemplo:** Um banco de dados distribuído precisa saber qual servidor tem permissão de *escrita primária* para evitar que dois servidores escrevam dados conflitantes ao mesmo tempo.

# Exemplo Conceitual

Considere um sistema de processamento de pedidos em uma loja virtual. Há cinco servidores trabalhando em conjunto:

P1   P2   P3   P4   P5

O processo **P5** é o coordenador.

- Ele decide qual servidor processará cada novo pedido.

Se **P5** falha, os demais processos precisam continuar operando.

- Nesse caso:

Quem deve assumir o papel de novo coordenador?

# Modelo básico do problema

Assumiremos que:

- cada processo tem um identificador único;
- os processos se comunicam por mensagens;
- o coordenador é um dos processos ativos;
- processos podem detectar ou suspeitar da falha do coordenador;
- após a eleição, todos os processos ativos devem reconhecer o mesmo líder.

Exemplo de identificadores:

P1 < P2 < P3 < P4 < P5

# Eleição em Sistemas Distribuídos

**Eleição** é o processo pelo qual os processos de um sistema distribuído escolhem um novo coordenador.

A eleição normalmente ocorre quando:

- o coordenador atual falha;
- novos processos entram no sistema;
- há suspeita de falha do líder;
- o sistema precisa reorganizar sua coordenação.
- A ideia central é simples: se o líder falhou, **alguém** precisa assumir o papel de líder.

# Por que isso é difícil em Sistemas Distribuídos?

Em sistemas distribuídos, não existe uma visão global perfeita do sistema. Cada processo conhece apenas informações locais e mensagens recebidas.

Dificuldades comuns:

- atrasos na rede;
- mensagens perdidas;
- processos que falham silenciosamente;
- ausência de relógio global;
- dificuldade de distinguir lentidão de falha real.

# Algoritmo de Eleição

É um conjunto de regras para que os nós concordem em quem deve ser o novo líder.

- O objetivo principal é a **Tolerância a Falhas (Fault Tolerance)**.
- Deve garantir que, após uma falha, *apenas um* nó seja eleito como líder.
- **Consenso:** O sistema precisa chegar ao consenso sobre a identidade do líder.
- **Exemplo:** Um grupo de servidores API Gateway. Se o servidor primário cair, os demais devem se comunicar e concordar rapidamente em qual deles assumirá o tráfego principal.

# Algoritmo de Eleição: Propriedades desejadas

Um algoritmo de eleição deve garantir:

- **Safety:** não deve haver dois coordenadores corretos ao mesmo tempo.
- **Vivacidade:** se houver processos ativos, algum coordenador deve ser eleito.
- **Consenso prático:** todos os processos ativos devem saber quem é o líder.
- **Eficiência:** a eleição deve usar poucas mensagens e terminar rapidamente.
- **Tolerância a Falhas:** Deve funcionar mesmo com nós falhando ou se comunicando mal.

# Visão Geral dos Algoritmos

Estudaremos dois algoritmos clássicos:

## 1. Algoritmo Bully

- baseado na escolha do processo ativo com maior identificador.
- processos com identificadores maiores “intimidam” os menores.

## 2. Algoritmo Ring

- processos organizados logicamente em anel.
- mensagens circulam pelo anel até definir o líder.

# Algoritmo "Bully" (Maioral)

Baseia-se na ideia de que o nó com **maior identificador** deve ser o líder.

Se um nó suspeita do líder, ele inicia uma votação (eleição).

O algoritmo usa três tipos principais de mensagem:

Mensagem	Significado
<b>ELECTION</b>	pergunta se há processo maior ativo
<b>OK</b>	indica que um processo maior está ativo
<b>COORDINATOR</b>	anuncia o novo coordenador

Na Foto: Lobo, o maioral (DC comics). Fonte: [DC.com](https://www.dc.com) 





# Algoritmo "Bully": Passo-a-Passo

Quando um processo  $P_i$  detecta falha do coordenador, ele inicia uma eleição:

- $P_i$  envia **ELECTION** para todos os processos com identificador maior.
- Se nenhum processo maior responder,  $P_i$  vence a eleição e se torna coordenador.
  - $P_i$  envia **COORDINATOR** para todos os processos menores.
- Se algum processo maior responder **OK**,  $P_i$  aguarda o resultado.
  - O processo maior inicia uma eleição até que o maior processo ativo seja eleito.
- O vencedor de uma eleição se anuncia como coordenador enviando uma mensagem para os demais processos

Note que a qualquer momento um processo pode receber uma mensagem de eleição de um processo de numeração menor. Um processo deve responder à mensagem para indicar que está ativo. Se um processo que estava previamente inativo volta à atividade, este pode iniciar uma eleição

# Algoritmo "Bully": Vantagens e Limitações

## Vantagens:

- Garante que o maior processo ativo seja eleito.
- Funciona bem quando os processos conhecem os demais participantes.
- Útil para ambientes com conjunto relativamente estável de processos.


## Limitações:

- Pode gerar muitas mensagens.
- Exige que cada processo conheça os identificadores dos outros processos.
- Depende de mecanismos de detecção de falha.
- Pode ser sensível a atrasos de rede.
- A entrada e saída dinâmica de processos pode tornar o controle mais difícil.

# Algoritmo Ring (Anel)

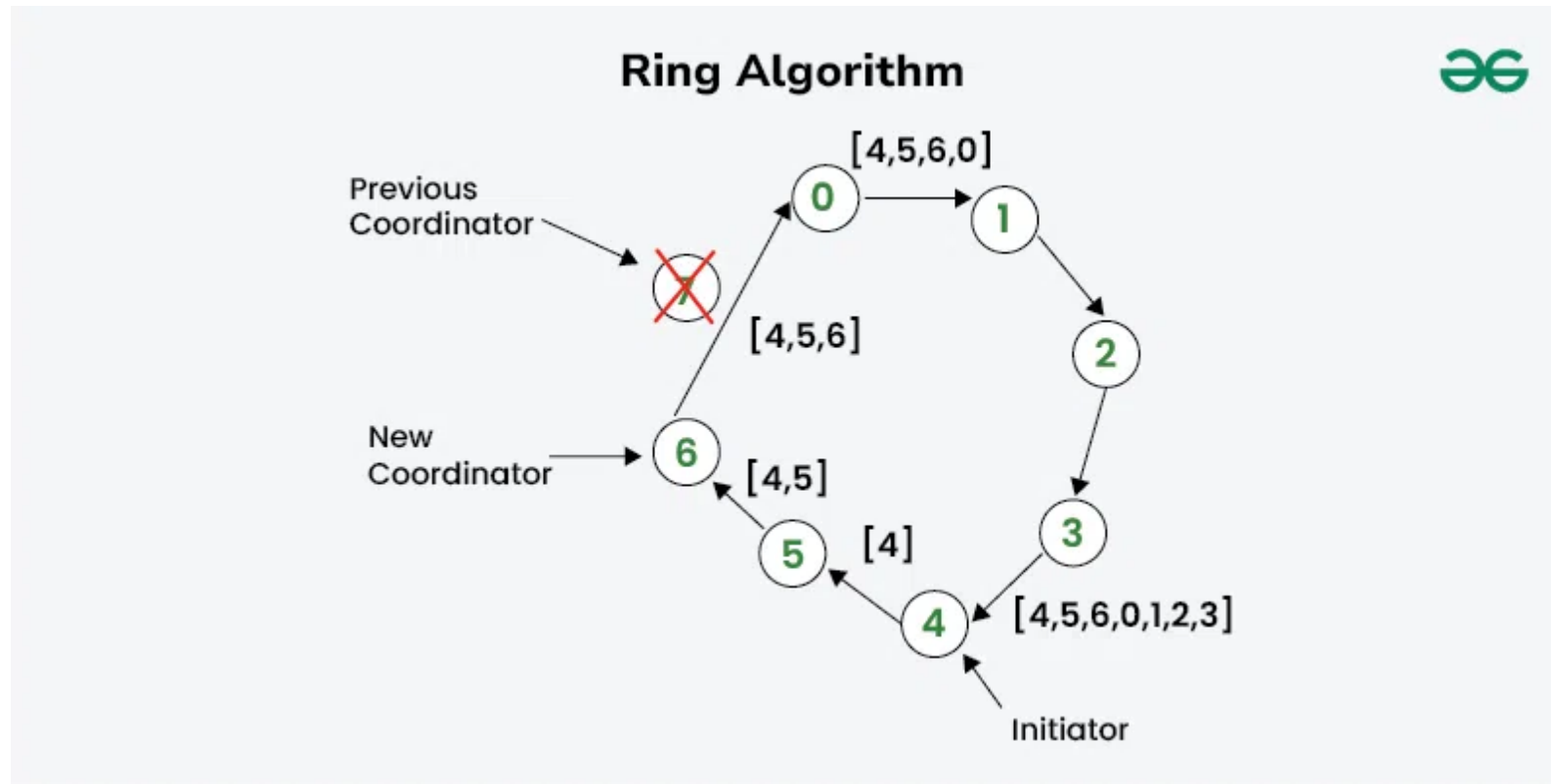
No algoritmo Ring, os processos são **ordenados** em um **anel lógico**.

- Cada processo conhece seu sucessor no anel.
- Exemplo:  $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_1$
- A mensagem de eleição circula pelo anel até retornar ao iniciador.

Na imagem: Anéis de poder elaborados por Sauron e forjados por Celebrimbor. Fonte: [The Lord of the Rings Wiki](#) .



# Algoritmo Ring: Exemplo



Note que as mensagens dos processos 1 e 2 estão omitidas no diagrama. Fonte da imagem: [GeekforGeeks.com](https://www.geekforgeeks.com/)

# Algoritmo Ring: Ideia Cental

Quando um processo  $P$  percebe que o coordenador não está mais ativo, este processo inicia uma eleição enviando uma mensagem de eleição **para seu sucessor**.

- A mensagem carrega os **identificadores** dos processos ativos.
- Se o próximo processo no anel não estiver ativo, a mensagem deve ser **enviada para o sucessor deste** ou para o próximo processo do anel que estiver ativo.
- A cada passo, o processo que **encaminha** a mensagem **adiciona seu próprio identificador** lógico à mensagem.
- Quando a mensagem **retornar ao processo que a enviou**, este a transforma em uma **mensagem de notificação**, informando o identificador lógico do processo eleito como coordenador e a nova configuração do anel.
  - Quando esta mensagem circular o anel, esta é então descartada.

# Mensagens no Ring

O algoritmo usa mensagens como:

Mensagem	Significado
<b>ELECTION</b>	circula pelo anel coletando IDs dos processos ativos
<b>COORDINATOR</b>	circula informando quem foi eleito

A eleição acontece por circulação de mensagens.

# Quando usar cada abordagem?

**Bully** pode ser mais adequado quando:

- os processos conhecem todos os demais;
- o grupo é pequeno ou moderado;
- deseja-se eleger rapidamente o maior processo ativo.

**Ring** pode ser mais adequado quando:

- os processos estão organizados logicamente em anel;
- deseja-se limitar o número de mensagens;
- cada processo deve conhecer poucos vizinhos.

# Problemas práticos em algoritmos de eleição

Na prática, algoritmos de eleição precisam lidar com:

- detecção imprecisa de falhas;
- mensagens duplicadas;
- processos que retornam após falha;
- particionamento de rede;
- eleições simultâneas;
- mudança dinâmica de participantes.

Esses problemas tornam a eleição mais complexa em sistemas reais.

# Exemplo prático de aplicação

Sistemas que podem precisar de eleição de líder:

- clusters de servidores;
- sistemas de replicação;
- bancos de dados distribuídos;
- sistemas de arquivos distribuídos;
- aplicações com coordenação de tarefas;
- serviços em nuvem com múltiplas instâncias.

A eleição ajuda o sistema a continuar funcionando mesmo quando o líder falha.

# Conclusão

Algoritmos de eleição são fundamentais para manter a coordenação em sistemas distribuídos.

Principais ideias da aula:

- sistemas distribuídos precisam lidar com falhas de coordenadores;
- eleição escolhe um novo líder entre os processos ativos;
- o algoritmo Bully escolhe o maior processo ativo por consultas diretas;
- o algoritmo Ring escolhe o maior processo ativo por circulação em anel;
- ambos dependem de comunicação por mensagens e detecção de falhas.

# Material Adicional

- Livro Coulouris: Capítulo 15, seções 15.3
- Livro Tanenbaum: Capítulo 6, seção 6.5
- [Richard John Anthony: Capítulo 6, seção 6.7](#) ↗
- [What is Leader Election in a Distributed System?](#) ↗
- [Leader Election Algorithms](#) ↗.

# Dúvidas e Discussão

# Questões para Estudo

---

1. Em poucas palavras, qual é o objetivo principal de um algoritmo de eleição em sistemas distribuídos?
2. Qual é o maior risco operacional que ocorre se dois nós diferentes acreditarem ser os líderes simultaneamente? (Dica: Pense em dados).
3. No Algoritmo Bully, qual tipo de nó deve iniciar a eleição quando detecta uma falha do líder?
4. Qual é a principal vantagem conceitual do Algoritmo Bully em relação à sua complexidade de implementação? E qual é sua desvantagem em redes muito grandes?

# Discussão

- Proponha uma solução que optimize o algoritmo Bully.
- Suponha que dois processos detectem a queda do coordenador simultaneamente e decidam por uma eleição usando o algoritmo Bully. Explique o que acontece.
- Considerando-se o algoritmo Ring, suponha que duas mensagens estejam circulando simultaneamente para a eleição de um novo coordenador. Proponha um algoritmo que elimine uma das mensagens.